

Network list

```
Administrator: Command Prompt

C:\WINDOWS\system32>docker network ls
NETWORK ID          NAME                DRIVER              SCOPE
3d339d32fd6a        bridge             bridge              local
789ee862add5        host               host                local
7213a87e7c25        none               null                local

C:\WINDOWS\system32>
```

Inspecting the bridge network

```
Administrator: Command Prompt

C:\WINDOWS\system32>docker network ls
NETWORK ID          NAME                DRIVER              SCOPE
3d339d32fd6a        bridge             bridge              local
789ee862add5        host               host                local
7213a87e7c25        none               null                local

C:\WINDOWS\system32>docker network inspect bridge
[
  {
    "Name": "bridge",
    "Id": "3d339d32fd6a41798f6483d2105fb2b992eaff30c53ceafd3431263e0d44e62e",
    "Created": "2021-05-13T04:47:14.7300618Z",
    "Scope": "local",
    "Driver": "bridge",
    "EnableIPv6": false,
    "IPAM": {
      "Driver": "default",
      "Options": null,
      "Config": [
        {
          "Subnet": "172.17.0.0/16",
          "Gateway": "172.17.0.1"
        }
      ]
    },
    "Internal": false,
    "Attachable": false,
    "Ingress": false,
    "ConfigFrom": {
      "Network": ""
    }
  }
]
```

Inspecting bridge while a ping container is running

```
Select Administrator: Command Prompt

    "Gateway": "172.17.0.1"
  }
]
},
"Internal": false,
"Attachable": false,
"Ingress": false,
"ConfigFrom": {
  "Network": ""
},
"ConfigOnly": false,
"Containers": {
  "823a1849eab0b98545b39762e7bddbe0751c4ec68ef478ffcd14f3541ef51e5a": {
    "Name": "dummy",
    "EndpointID": "afd995c7896d30a253ea7e7b6a825affdf14cda5648b2699275171fbe8d3ec7c",
    "MacAddress": "02:42:ac:11:00:02",
    "IPv4Address": "172.17.0.2/16",
    "IPv6Address": ""
  }
},
"Options": {
  "com.docker.network.bridge.default_bridge": "true",
  "com.docker.network.bridge.enable_icc": "true",
  "com.docker.network.bridge.enable_ip_masquerade": "true",
  "com.docker.network.bridge.host_binding_ipv4": "0.0.0.0",
  "com.docker.network.bridge.name": "docker0",
  "com.docker.network.driver.mtu": "1500"
},
"Labels": {}
}
```

Another ping container set to ping container dummy (using the IP address)

```
Administrator: Command Prompt

C:\WINDOWS\system32>docker run --rm -d -e PING_TARGET=172.17.0.2 --name pinger blueaxis/ping
58c66efd21ad086e5401bda0eca9d830505c7f21d97008922703002723087c73

C:\WINDOWS\system32>docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS   NAMES
58c66efd21ad   blueaxis/ping  "sh -c 'ping $PING_T..." 13 seconds ago Up 9 seconds           pinger
823a1849eab0   blueaxis/ping  "sh -c 'ping $PING_T..." 5 minutes ago  Up 5 minutes           dummy

C:\WINDOWS\system32>docker logs pinger
PING 172.17.0.2 (172.17.0.2) 56(84) bytes of data.
64 bytes from 172.17.0.2: icmp_seq=1 ttl=64 time=0.053 ms
64 bytes from 172.17.0.2: icmp_seq=2 ttl=64 time=0.059 ms
64 bytes from 172.17.0.2: icmp_seq=3 ttl=64 time=0.088 ms
64 bytes from 172.17.0.2: icmp_seq=4 ttl=64 time=0.084 ms
64 bytes from 172.17.0.2: icmp_seq=5 ttl=64 time=0.082 ms
64 bytes from 172.17.0.2: icmp_seq=6 ttl=64 time=0.081 ms
64 bytes from 172.17.0.2: icmp_seq=7 ttl=64 time=0.055 ms
64 bytes from 172.17.0.2: icmp_seq=8 ttl=64 time=0.087 ms
64 bytes from 172.17.0.2: icmp_seq=9 ttl=64 time=0.088 ms
64 bytes from 172.17.0.2: icmp_seq=10 ttl=64 time=0.036 ms
64 bytes from 172.17.0.2: icmp_seq=11 ttl=64 time=0.080 ms
64 bytes from 172.17.0.2: icmp_seq=12 ttl=64 time=0.083 ms
64 bytes from 172.17.0.2: icmp_seq=13 ttl=64 time=0.082 ms
64 bytes from 172.17.0.2: icmp_seq=14 ttl=64 time=0.095 ms
64 bytes from 172.17.0.2: icmp_seq=15 ttl=64 time=0.079 ms
64 bytes from 172.17.0.2: icmp_seq=16 ttl=64 time=0.079 ms
64 bytes from 172.17.0.2: icmp_seq=17 ttl=64 time=0.082 ms
64 bytes from 172.17.0.2: icmp_seq=18 ttl=64 time=0.021 ms
64 bytes from 172.17.0.2: icmp_seq=19 ttl=64 time=0.032 ms
```

Another ping container set to ping container dummy (using container name), does not work

```
Administrator: Command Prompt

C:\WINDOWS\system32>docker run --rm -d -e PING_TARGET=dummy --name pinger blueaxis/ping
ad149194c2e474d41794dda4a2dc03fb4cd40bbe864e83be211b841d2a767e39

C:\WINDOWS\system32>docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS          NAMES
823a1849eab0   blueaxis/ping  "sh -c 'ping $PING_T..."  12 minutes ago Up 11 minutes           dummy

C:\WINDOWS\system32>
```

Running common commands on a custom network

```
Administrator: Command Prompt

C:\WINDOWS\system32>docker network create skynet
b0ca697fc3ff4d48964a7c466ddc31c2c240062166337209a4088c13ecace7dd

C:\WINDOWS\system32>docker network ls
NETWORK ID      NAME      DRIVER  SCOPE
3d339d32fd6a    bridge    bridge  local
789ee862add5    host      host    local
7213a87e7c25    none      null    local
b0ca697fc3ff    skynet    bridge  local

C:\WINDOWS\system32>docker network inspect skynet
[
  {
    "Name": "skynet",
    "Id": "b0ca697fc3ff4d48964a7c466ddc31c2c240062166337209a4088c13ecace7dd",
    "Created": "2021-05-13T08:32:16.4483146Z",
    "Scope": "local",
    "Driver": "bridge",
    "EnableIPv6": false,
    "IPAM": {
      "Driver": "default",
      "Options": {},
      "Config": [
        {
          "Subnet": "172.18.0.0/16",
          "Gateway": "172.18.0.1"
        }
      ]
    }
  }
],
```

Adding another ping container set to ping container dummy (using container name). This time, the host name resolves to the IP address of dummy

```
Administrator: Command Prompt
C:\WINDOWS\system32>docker run --rm -d --network skynet --name dummy blueaxis/ping
29250f9c7d92428d0b8831e26712f7939254c638f9fe4d2167b90cdf370eb254

C:\WINDOWS\system32>docker run --rm -d --network skynet --name dummy blueaxis/ping
docker: Error response from daemon: Conflict. The container name "/dummy" is already in use by container "29250f9c7d92428d0b8831e26712f7939254c638f9fe4d2167b90cdf370eb254". You have to remove (or rename) that container to be able to reuse that name.
See 'docker run --help'.

C:\WINDOWS\system32>docker run --rm -d --network skynet -e PING_TARGET=dummy --name pinger blueaxis/ping
0d0f639afae27b5601637fb6e82aa5630d928d973ba5fd234329e703abd8030d

C:\WINDOWS\system32>docker logs pinger
PING dummy (172.18.0.2) 56(84) bytes of data.
64 bytes from dummy.skynet (172.18.0.2): icmp_seq=1 ttl=64 time=0.062 ms
64 bytes from dummy.skynet (172.18.0.2): icmp_seq=2 ttl=64 time=0.104 ms
64 bytes from dummy.skynet (172.18.0.2): icmp_seq=3 ttl=64 time=0.111 ms
64 bytes from dummy.skynet (172.18.0.2): icmp_seq=4 ttl=64 time=0.038 ms
64 bytes from dummy.skynet (172.18.0.2): icmp_seq=5 ttl=64 time=0.104 ms
64 bytes from dummy.skynet (172.18.0.2): icmp_seq=6 ttl=64 time=0.145 ms
64 bytes from dummy.skynet (172.18.0.2): icmp_seq=7 ttl=64 time=0.076 ms
64 bytes from dummy.skynet (172.18.0.2): icmp_seq=8 ttl=64 time=0.042 ms
64 bytes from dummy.skynet (172.18.0.2): icmp_seq=9 ttl=64 time=0.038 ms
64 bytes from dummy.skynet (172.18.0.2): icmp_seq=10 ttl=64 time=0.106 ms
64 bytes from dummy.skynet (172.18.0.2): icmp_seq=11 ttl=64 time=0.060 ms
64 bytes from dummy.skynet (172.18.0.2): icmp_seq=12 ttl=64 time=0.117 ms
64 bytes from dummy.skynet (172.18.0.2): icmp_seq=13 ttl=64 time=0.056 ms
64 bytes from dummy.skynet (172.18.0.2): icmp_seq=14 ttl=64 time=0.048 ms
64 bytes from dummy.skynet (172.18.0.2): icmp_seq=15 ttl=64 time=0.111 ms
64 bytes from dummy.skynet (172.18.0.2): icmp_seq=16 ttl=64 time=0.083 ms
```

Connecting between containers in a network. I was stuck in here for a while. Running `docker ps` returns an empty list at first. It turns out that I have to specify `POSTGRES_PASSWORD` to a non-empty value.

```
Administrator: Command Prompt
C:\WINDOWS\system32>docker network create skynet
Error response from daemon: network with name skynet already exists

C:\WINDOWS\system32>docker run --rm -d -e POSTGRES_PASSWORD=password --name widgetdb --network skynet -p 5432 postgres
828c488183f735ddfbc8b8adc8710150cc89fb05f55809faee4dd3ce5133f00

C:\WINDOWS\system32>docker run --rm -d -e POSTGRES_PASSWORD=password --name gadgetdb --network skynet -p 5432 postgres
67b39a1c4600169b7111f19db7308c84aac5be868eb20d3587a0a2b2180997c5

C:\WINDOWS\system32>docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS                               NAMES
67b39a1c4600   postgres  "docker-entrypoint.s..." 5 seconds ago  Up 2 seconds  0.0.0.0:53065->5432/tcp             gadgetdb
828c488183f7   postgres  "docker-entrypoint.s..." 29 seconds ago  Up 27 seconds  0.0.0.0:53062->5432/tcp             widgetdb

C:\WINDOWS\system32>docker exec -it widgetdb /bin/bash
root@828c488183f7:/# psql -U postgres
psql (13.2 (Debian 13.2-1.pgdg100+1))
Type "help" for help.

postgres=# \q
root@828c488183f7:/# psql -U postgres -h gadgetdb
Password for user postgres:
psql (13.2 (Debian 13.2-1.pgdg100+1))
Type "help" for help.

postgres=# \q
root@828c488183f7:/# exit
exit

C:\WINDOWS\system32>
```

Binding ports to the host

```
Administrator: Command Prompt

C:\WINDOWS\system32>docker run --rm -d -e POSTGRES_PASSWORD=password --name widgetdb --network skynet -p 5432:5432 postgres
6c18f0456156753befea5b247941f5dac0f4c117718d2ebb40e32c2b596e43be

C:\WINDOWS\system32>docker exec -it widgetdb /bin/bash
root@6c18f0456156:/# psql -U postgres -h localhost
psql (13.2 (Debian 13.2-1.pgdg100+1))
Type "help" for help.

postgres=# \q
root@6c18f0456156:/# exit
exit

C:\WINDOWS\system32>
```